

CFA based SBOX and Modified Mixcolumn Implementation of 8 Bit Datapath for AES

K. Rahimunnisa¹ S.Sureshkumar² J.Jayakumar³ Jishamol T. K⁴

^{#1,4} Karunya University/ECE Department, Coimbatore, India.

Email: krahimunnisa@gmail.com

^{#2} Dr.N.G.P Institute of Technology/ECE Department, Coimbatore, India.

^{#3} Karunya University/EEE Department, Coimbatore, India.

Abstract—Secure data transmission is very important in any communication systems. Network Security provides many techniques for efficient data transmission through unprotected network. Cryptography provides a method for securing the transmission of information by the process of encryption. Encryption converts the message in to unreadable form (Cipher Text) . Decryption converts this Cipher Text back to original message. Advanced Encryption Standard (AES) has been used as the first choice of cryptographic algorithm for many security based applications because of the high level of security and flexibility of implementation in hardware and software. This paper presents an area efficient, low power design for AES based on an 8-bit data path making it suitable for wireless security applications. It has a significant power-area-latency performance improvements over other existing AES designs. For high performance applications, AES S-box and inverse S-box implemented using composite field Arithmetic (CFA). Also low resource Mixcolumn structure is used in this structure. The 8 bit data path architecture is implemented in XILINX 13.2 and simulated using MODELSIM 6.5 software. Also the power and area calculation is done with the help of SYNOPSYS software.

Index Terms— Advanced Encryption Standard (AES), S-box, Composite Field Arithmetic, FPGA, ASIC.

I. INTRODUCTION

The Advanced Encryption Standard (AES) has been accepted by National Institute standards and Technology (NIST) as the symmetric key standard for encryption and decryption [1]. In encryption, AES operates on 128 bit plaintext input and a key that can be specified to be 128, 192 or 256 bits to generate the ciphertext. In the AES-128, the ciphertext is generated after 10 rounds, where each encryption round (except for the final round) consists of four transformations. The four transformations include SubBytes, ShiftRows, MixColumns, and AddRoundKey. To obtain the original data from the ciphertext, the AES decryption algorithm is utilized. The transformations in decryption are the reverse of encryption ones [1].

There is a continued demand for better hardware AES implementations, especially in resource-critical environments. This paper aims to support the low-resource end of the design space where minimization of power, area and latency is needed. AES designs typically based on 128 bit data path, have been widely presented. Many of the attempts at Application Specific Integrated Circuit (ASIC) designs for the AES considered only a 32-bit data path AES design as the minimum [2]. In previous designs [3], the author explored the Application Specific Instruction Processor (ASIP) for a Field-Programmable Gate Array

(FPGA), which utilized an 8-bit data path for AES. The schemes in [5-8] present suitable structure of CFA for the S-box and inverse S-box implementation. In this paper, an area efficient, low power AES design using 8-bit data path is explained for the implementation of the AES, which provides significant improvements over the previous AES designs. This design utilizes an 8-bit data path between key and state processing, with resource sharing of the SubByte operation.

The remainder of this paper is organized as follows. In section II, AES S-box and inverse S-box using CFA are explained. Section III explains the design of 8 bit data path AES using CFA. The results and discussions are given in Section IV. The paper ends by drawing some conclusions in Section V.

II. AES S-BLOCK AND INVERSE S-BLOCK USING CFA

Generally, Sbox and Inverse Sbox are implemented using LUTs as they are predefined and precomputed values. But this technique occupies more area. Hence a Composite Field Arithmetic (CFA) based implementation is used to reduce area.

In this section, we describe the S-box and inverse S-box operations and their composite field realizations. The S-box and inverse S-box are nonlinear operations which take 8-bit as input and generate 8-bit as outputs. In the S-box, the irreducible polynomial of $P(x) = x^8 + x^4 + x^3 + x + 1$ is used to construct the Galois field $GF(2^8)$. The S-box consists of the multiplicative inversion, followed by an affine transformation[1]. The inverse S-box consists of an inverse affine transformation followed by multiplicative inversion. The composite fields can be represented by using normal basis or polynomial basis. The S-box and inverse S-box for using composite fields and polynomial basis are shown in Fig 1. The block 1 consists of transformation matrix and modulo addition operation, block 2 consists of squarer, scaling, modulo multiplication and modulo addition operations, block 3 consists of multiplicative inversion, block 4 consists of multiplication operations and block 5 consists of mixed inverse and affine transformations as in Fig 1 [9]. As in figure, for the S-box, the transformation matrix transforms a field element X in the binary field $GF(2^8)$ to the corresponding representation in the composite fields $GF(2^8)$ or $GF(2^4)$.

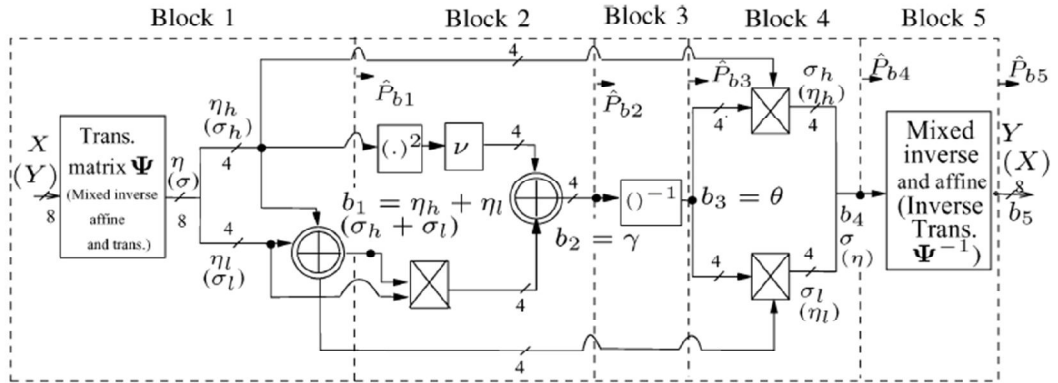


Fig 1 The S-box (the inverse S-box) using composite field arithmetic (CFA) and polynomial basis

Let y_1 and y_h are the field elements in the sub field $GF(2^4)$. The decomposition can be further applied to represent $GF(2^4)$ as a linear polynomial over $GF(2^2)$ and then the $GF(2)$. The multiplicative inversion consists of multiplications the modulo-2 additions, squaring and an inversion in the sub-field $GF(2^4)$ over $GF(2)$. The implementation of inversion in $GF(2^4)$ is shown below in Fig 2. The Fig 2.(a) shows square multiply approach and Fig 2.(b) shows multiple decomposition approach. In Fig. 1, the modulo-2 additions, consisting of four XOR gates, are shown by two concentric circles with a plus inside. Furthermore, the multiplication operations in $GF(2^4)$ are shown by rectangles with crosses inside. It is also noted that the implementation of the multiplicative inversion can be performed using the field represented by $GF((2^4))^2$ or the field represented by $GF(((2^2))^2)^2$. After calculating the inversion in the composite field, affine transformation is used to transform the composite field representation to the field element Y [4]. Also the individual blocks in the S-box architecture is shown in Fig 3.

III. 8 BIT DATA PATH AES DESIGN

The aim of this design is to minimize the performance dimensions power, area and latency. Such minimization is achieved by the use of appropriate resource sharing, simple compact memory architecture, field arithmetic optimization, avoiding unnecessary switching activity, adopting an 8-bit data path width, and minimizing memory transfer. Typically, CMOS power consumption is usually dominated by the dynamic power consumption from the switching activity of the device, and the static power is neglected. In previous AES papers [3] frequently uses generate much unwanted dynamic switching activity due to path length differences.

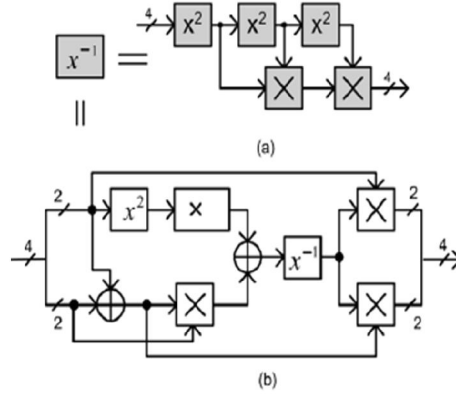


Fig 2 Implementation of inversion in $GF(2^4)$ (a) square multiply approach (b) multiple decomposition approach

In this design, this is checked by the placement of the shift register between the SubBytes and MixColumns operators together with the clock-gate-style enabled signals to prevent unwanted activity in both Key Expansion and state-processing parts of the data path.

In this design first the key is fed into the key memory. Next, as the plaintext is supplied, the first AES round is processed (simply Add Round Key), and the results are stored in the data memory. The middle round processing proceeds in column order. The final round is same as the middle rounds in AES except that Mix Columns is bypassed and the result bytes are stored in the output register. In this design, two 16 byte memories are used for storing 128 bits of data and key. First memory is for the Round Key and the second for the state.

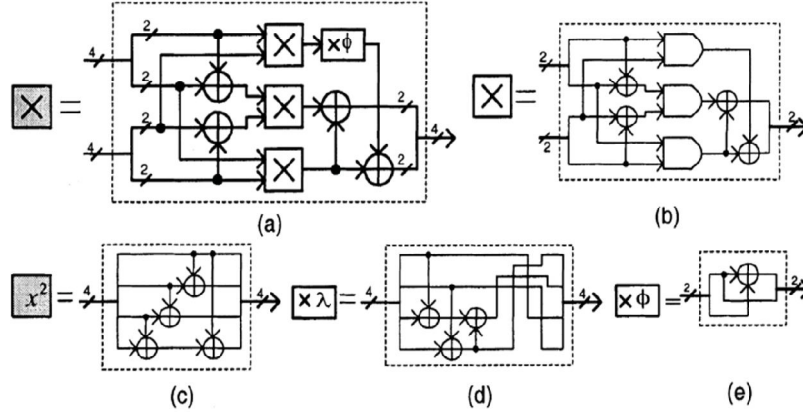


Fig 3 Implementation of individual blocks in CFA: (a) multiplier in $GF(2^4)$ (b) multiplier in $GF(2^2)$ (c) squarer in $GF(2^4)$ (d), (e) constant multiplier

The block diagram of 8-bit data path AES encryption is shown in Fig 4. This involves both key expansion and Encryption, both are taking place simultaneously. AES defines a 16×16 matrix of byte values, called an S-box. The S-box and inverse S-box can be implemented using either look up table (LUT) or CFA. Initially S-box and the inverse S-box is implemented by LUT approach. But this approach may not be preferable for

8 bit data path encryption decryption combined AES Cipher waveform using CFA and low resource mixcolumn. The device used was Virtex 6.

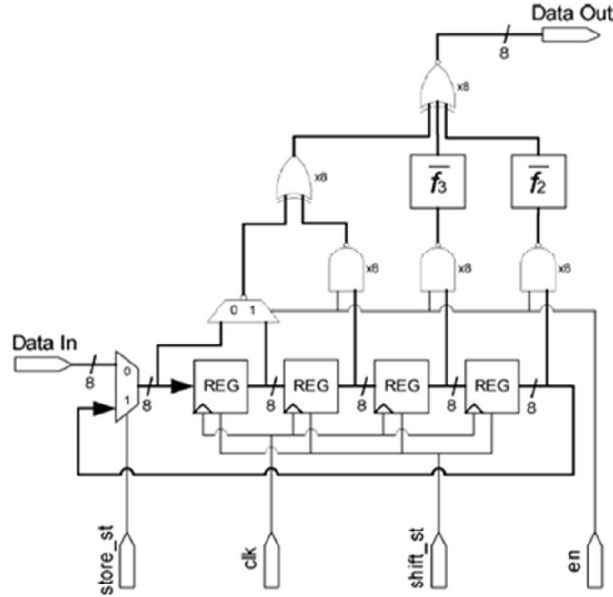


Fig 5 Low Resource MixColumn Circuit

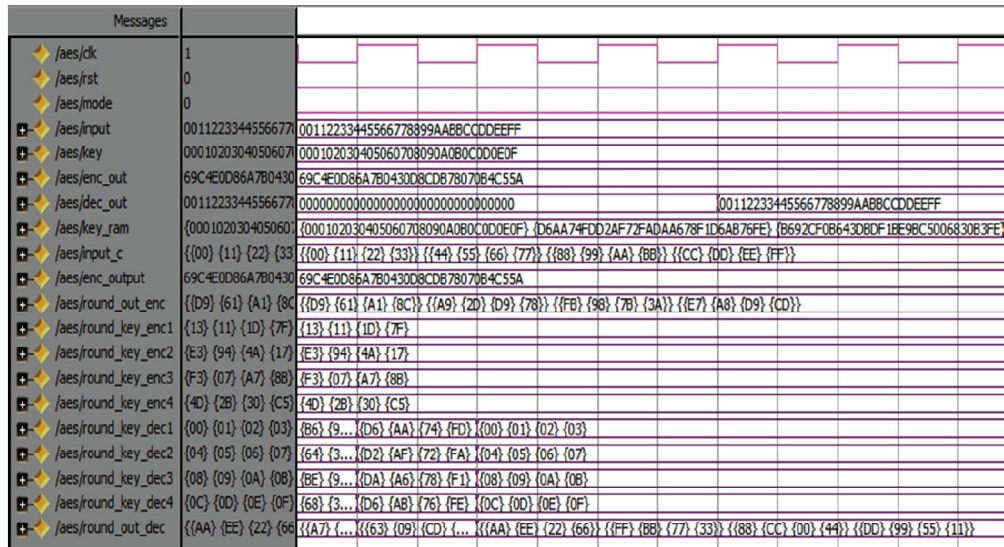


Fig. 6 8 bit AES cipher waveform using CFA

Next graphs show the FPGA performance comparison between 128 bit and 8 bit data path AES cipher. Fig. 7 shows the area comparison between 128 bit and 8 bit data path AES cipher. The 8-bit data path AES uses less number of RAMs, Registers, Multiplexers and XORs, thus it reduces the area as compared to 128-bit data path AES. Fig. 8 shows the clock cycles for both 128-bit and 8-bit data path AES cipher. The number of clock cycles for 8 bit data path AES encryption and decryption is less compared to 128 bit data path AES encryption and decryption. Fig 9 gives the area comparison between AES using LUT and AES using CFA cipher. The AES cipher using CFA reduces the use RAMs in LUT implementation by increasing the XOR operations.

Table 1 shows the ASIC performance comparison between 8 bit data path AES architecture using CFA and low resource mixcolumn with the existing architectures. SYNOPSIS software was used to determine the power and area. 90 nm technology was used and compared with the results available in literature.

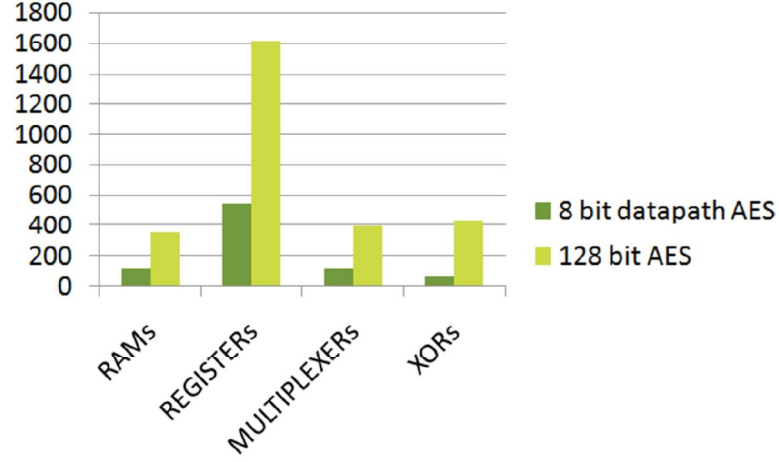


Fig. 7 Area comparison between 128 and 8 bit data path AES

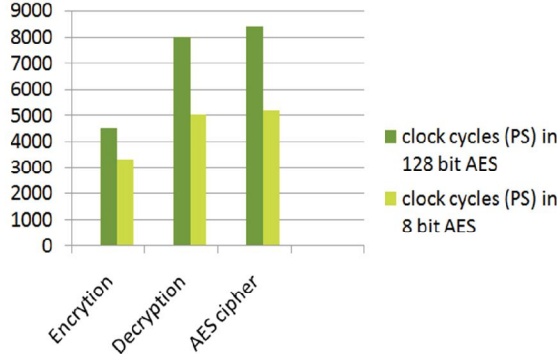


Fig. 8 Comparison between clock cycles in 128 and 8 bit data path AES

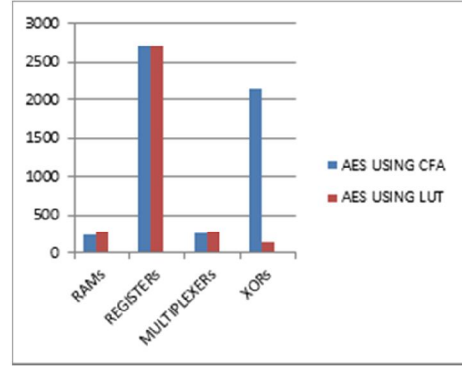


Fig.9 Area comparison between 8 bit data path AES using LUT and CFA

V. Conclusion

This paper presents the FPGA and ASIC implementation of low power and low area for the AES cipher using 8 bit data path. The S-box and inverse S-box can be implemented using either look up table (LUT) or CFA. Initially S-box and the inverse S - box is implemented by LUT approach. But this approach may not be preferable for high performance AES implementations because, it requires a large number of registers for storing the 256 bytes. So the modification has been implemented by replacing the LUT approach with CFA. This reduces the number of registers in LUT approach. Also a low resource MixColumn circuit is used for reducing the resources and thus improves performance of 8 bit data path AES. The 8 bit data path Design for Advanced Encryption Standard (AES) is coded in Very High Speed Integrated Circuit Hardware Description Language (VHDL). The synthesis is done by Xilinx ISE 13.2i and simulation is performed by ModelSim XE III 6.3 simulator. The device Virtex 7 XC7V585TL is targeted device for FPGA implementation. The FPGA implementation of 8 bit data path AES using CFA gives the power consumption of 554 mw for 8 bit data path AES architecture. The ASIC implementation of the proposed 8 bit data path AES architecture using CFA achieves power of 693.35 uW and an area of 28276 nm² using the above area and power reduction techniques. This proves that the 8 bit data path AES architecture using CFA reduces the power and area than all existing architectures. Compared to existing designs, this shows the best P-A-L (power, area, latency) efficiency. It is hoped that this low resource design for AES will open up new opportunities for the AES in

resource- sensitive applications. This design focuses on power, area and latency and not on the speed. Hence, the future work can be concentrated on the speed, since the speed is essential for the real time applications.

TABLE 1 ASIC PERFORMANCE COMPARISON OF 8 BIT DATA PATH AES ARCHITECTURE USING CFA WITH THE EXISTING ARCHITECTURE

AUTHOR	ARCHITECTURE	TECHNOLOGY	POWER	AREA
S.Y. Lin (2007)	NORMAL AES [10]	0.13 um	40.9 mw	86.2 K gates
Al Wen Luo et.al (2011)	PIPELINED AES [11]	0.18 um	14.025mW	52131.166um square
Alma'aitah. A (2010)	SUBPIPELINED AES [12]	0.18um	84.6mW	-
Choi.H.S (2008)	PARALLEL AES [13]	0.90um	7.56mW	-
This Work	8 BIT DATA PATH AES USING LUT	0.90um	1628.91uW	70178nm square
This Work	8 BIT DATA PATH AES USING CFA	0.90um	693.35uW	28276 nm square

REFERENCES

- [1] Nat. Inst. Standards Technol. (NIST), 2001 "Federal Information Processing Standards (FIPS) Publication 197," Advanced Encryption Standard, Nov. 2001.
- [2] Avi kak "AES: The Advanced Encryption Standard" Lecture Notes on "Computer and Network Security", Feb 26, 2013.
- [3] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," in Proc. ASIACRYPT, Gold Coast, Qld., Australia, Dec. 2001.
- [4] H. Kuo, I. Verbaauwhede, and P. Schaumont, "A 2.29 Gbits/Sec, 56Mw non-pipelined Rijndael AES encryption IC in a 1.8 V 0.18 um CMOS technology," in Proc. CICC, Orlando, FL, pp. 147–150, 2002.
- [5] Tim Good and Mohammed Benaissa "692-nW Advanced Encryption Standard (AES) on a 0.13- m CMOS" in IEEE Transactions on Very Large Scale Integration (VLSI) systems, vol. 18, no. 12, December 2010.
- [6] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES implementation on a grain of sand," Proc. Inst. Electr. Eng. Inf. Security, vol. 1, pp. 13–20, 2005.
- [7] D. Canright, "A very compact S-box for AES," in Proc. CHES Edinburgh, U.K. Vol. 3659, LNCS, pp. 441–456, 2005
- [8] C. Paar, "Efficient VLSI architectures for bit-parallel computation in Galois fields,," Ph.D. dissertation, Inst. Exp. Math., Univ. Essen, Essen, Germany, Jun. 1994.
- [9] Xinniao Zhang, Member, Keshab K. Parhi, "On the Optimum Constructions of Composite Field for the AES Algorithm", IEEE Transactions on circuits and systems II: express briefs, vol. 53, no. 10, October 2006.
- [10] S.Y. Lin and C.T. Huang, "A high-throughput low-power AES cipher for network applications," in Proc. ASP-DAC, Yokohama, Japan, pp. 595–600, 2007.
- [11] Luo. A. W, Qing Ming Yi and Min Shi, "Design and implementation of Area optimized AES based on FPGA", International Conference on Business Management and Electronic Information, pp 743-746, 2011.
- [12] Alma'aitah. A and Zine-Eddine Abid, "Area Efficient High Throughput subpipelined Design of the AES in CMOS 180 nm", 5 the International Conference on Design and Test Workshop, pp 31-36, 2010.
- [13] Choi.H.S, Joong H.Choi and Jong Tae Kim, "Low Power AES Design using Parallel Architecture", International Conference on Convergence and Hybrid Information Technology, pp 413-417, 2008.